```
procedimento HIPOT (HIP, CAT1, CAT2);
        real: DIP, CAT1, CAT2;
        início
        | HIP ← RAIZ (CAT1 ** 2 + CAT2 ** 2);
        fim; {HIPOT}
```

Poderemos utilizar este procedimento duas vezes. Na primeira, para calcular a diagonal (*L*) do retângulo do plano da base do paralelepípedo e na segunda para calcular a diagonal (*D*) do paralelepípedo:

```
início
|  procedimento HIPOT (HIP, CAT1, CAT2)
|     real: HIP, CAT1, CAT2;
|     início
|     | HIP ← RAIZ (CAT1 ** 2 + CAT2 ** 2);
|     fim; {HIPOT}
|     real: A, B, C; {lados do paralelepípedo}
|     real: L, D;
|     leia (A, B, C);
|     HIPOT (L, A, B);     {L = √(A² + B²)}
|     HIPOT (D, L, C);     {D = √(L² + C²)}
|     imprima (D);
fim.
```

Exemplo 2:

Resolver o problema anterior utilizando uma função ao invés de um procedimento.

```
início
|  função HIP (CAT1, CAT2): real;
|     real: CAT1, CAT2;
|     início
|     | HIP ← RAIZ (CAT1 ** 2 + CAT2 ** 2);
|     fim; {HIP}
|  real: A, B, C;
|  leia (A, B, C)
|  imprima (HIP (HIP (A, B), C));
fim.
```

Exemplo 3:

Dado o procedimento DIGVER abaixo, escrever um algoritmo para:

a) Dado um número de cinco algarismos e o dígito verificador associado, verificar se o dígito está correto.

b) Gerar os dígitos verificadores dos números da seqüência: 1 a 9999.


```
procedimento DIGVER (NÚMERO, DIG);
   inteiro: NÚMERO;
   caracter: DIG;
   início
   |  inteiro: D1, D2, D3, D4, D5, SOMA, RESTO, COMP;
   |  "DECOMPOR NÚMERO NOS SEUS DÍGITOS CONSTITUINTES LANÇANDO-OS
   |  EM D1, D2, D3, D4 e D5";
   |  SOMA ← D1 * 6 + D2 * 5 + D3 * 4 + D4 * 3 + D5 * 2;
   |  RESTO ← SOMA mod 11;
   |  COMP ← 11 − RESTO;
   |  se COMP = 11
   |  |     então DIG: ← "0";
   |  |     senão
   |  |        se COMP = 10
   |  |        |  então DIG ← "X";
   |  |        |  senão DIG ← COMP;
   |  |        fim se;
   |  fim se;
fim {DIGVER};
```

```
início
|  procedimento DIGVER (NÚMERO, DIG);
|     .
|     .
|     .
|     .
|     .
|  fim; {DIGVER}
|  inteiro: NUM,            {número dado}
|           DIGITODADO     {dígito do número dado}
|           DIGITOGER      {dígito gerado}
|           I;             {número da seqüência}
|  leia (NUM, DÍGITO);
|  DIGVER (NUM, DIGITOGER);
|  se DIGITODADO = DIGITOGER
|  |  então imprima ("DÍGITO CORRETO");
|  |  senão imprima ("DÍGITO INCORRETO");
|  fim se;
|  para I de 1 até 9999 faça
|  |     DIGVER (I, DIGITOGER);
|  |     imprima (I, "–", DIGITOGER);
|  fim para;
fim.
```

Exemplo 4:

Escreva um algoritmo, em PORTUGOL, para um programa que calcule e imprima o valor de *S*, dado pela expressão a seguir: